



Distance Learning System



# Uvod u Python

Python and programming fundamentals

# Šta je Python

---



- **Python** je programski jezik
- Izmišljen je 1991 godine
- Autor Python-a je **Guido Van Rossum**
- Jezik **otvorenog koda**
- Trenutna verzija je verzija **3**
- Veoma zastupljen u web aplikacijama



# Programski jezik Python

---



- Python je **jezik visokog nivoa**
- Python se **najčešće interpretira**
- Python ima **jednostavnu sintaksu**
- Python je **hibridan** jezik
- Python je najzastupljeniji u **web** i **data science** aplikacijama
- Python je podržan na gotovo svim platformama

# Izvori za učenje

---

- <https://docs.python.org/3/library/index.html>
- <https://www.geeksforgeeks.org/python-programming-language/>

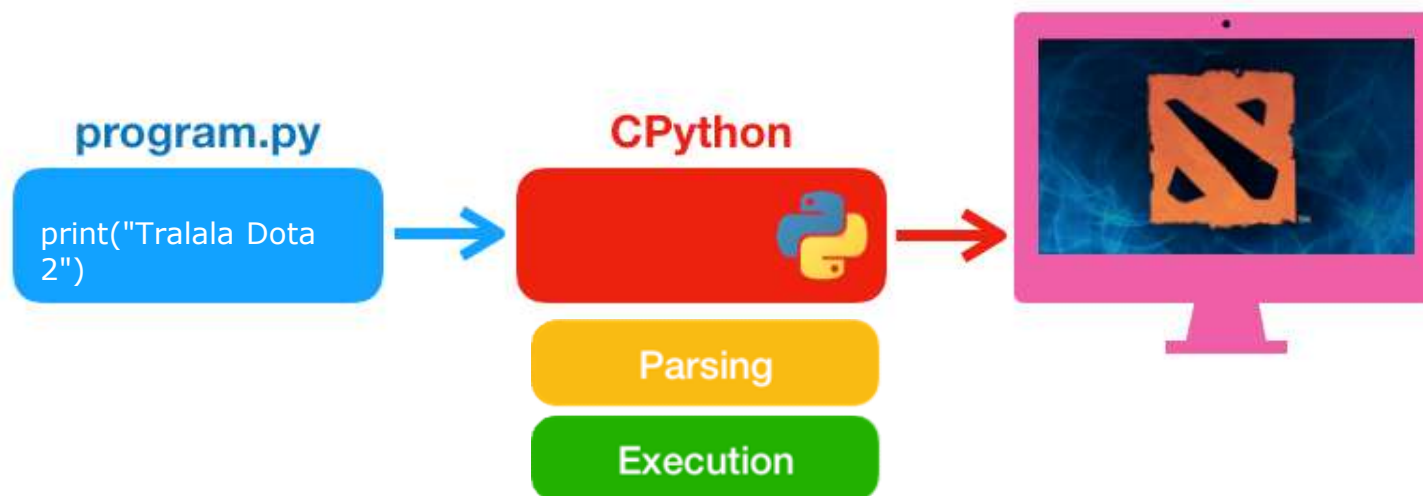
# Kako funkcioniše Python program?

---

- Život Python programa zavisi od implementacije
- Trenutno postoji mnoštvo implementacija Python-a (Jython, PyPy, Anaconda)
- Najzastupljenija implementacija Python-a je **CPython**



# Kako funkcioniše (C)Python program?



- Python program, u CPython implementaciji započinje **izvornim kodom** u tekstualnom formatu
- Kod se zatim prosleđuje **interpreteru**, koji ga **analizira**, **prevodi** i **startuje**

# Kako ja da startujem Python program?

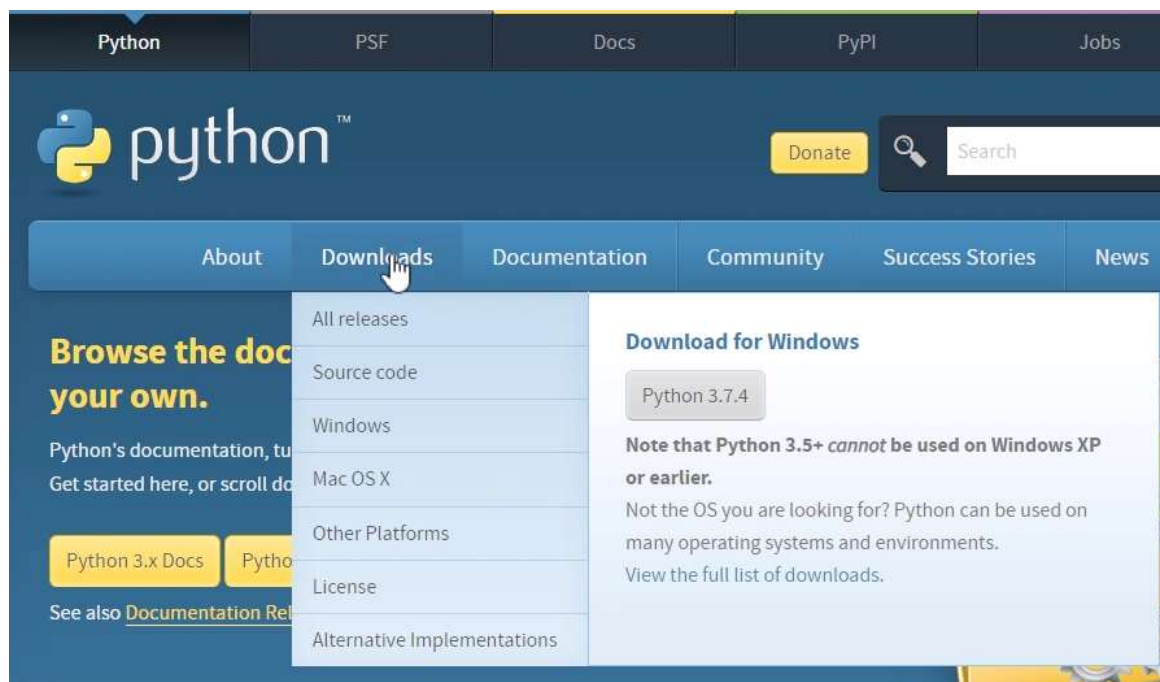
---

- Da bi Python program bio startovan, potrebno je imati:
  - Python **program**
  - Python **interpreter**

# Preuzimanje Python interpretera

<https://www.python.org/>

- Python je na nekim operativnim sistemima podrazumevano instaliran dok se na nekima mora preuzeti
- Proverite da li je Python već instaliran konzolnom komandom:  
`python -V`  
Ili:  
`python3 -V`
- Ukoliko python nije instaliran, preuzmite ga sa adrese desno





# Preuzimanje Python interpretera

---

Python **nije** instaliran



```
Command Prompt
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vpc>python
'python' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\vpc>
```

Python **je** instaliran

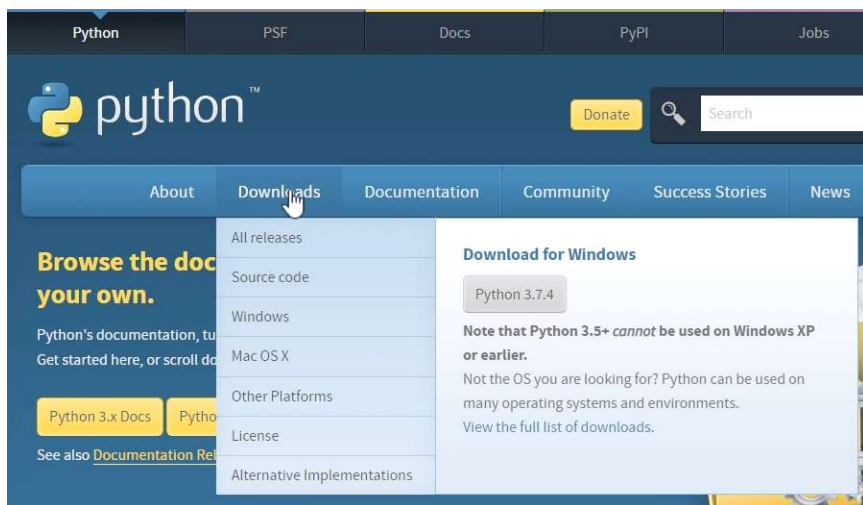


```
Command Prompt
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vpc>python -V
Python 3.7.4

C:\Users\vpc>
```

# Preuzimanje Python interpretera



Prilikom preuzimanja, **ne zaboravite** da dodate python u sistemsku putanju

Odaberite verziju koja odgovara verziji Vašeg sistema, preuzmite i pokrenite instalaciju



# Python, interaktivni mod

---

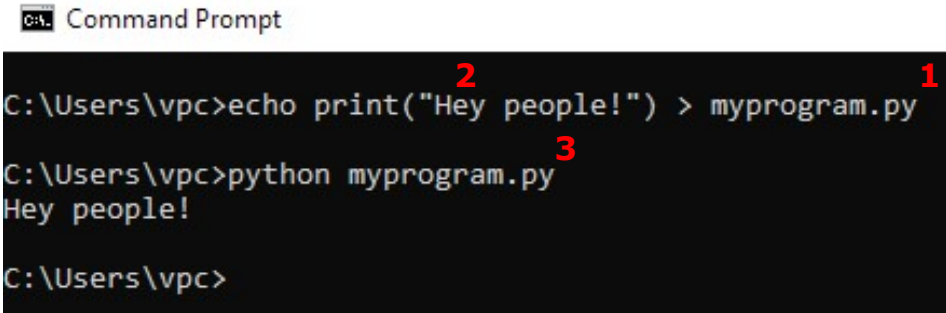
- Kucanjem komande **python** u konzoli, startuje se Python interaktivni mod
- Ovaj mod podrazumeva direktno kucanje komandi Python jezika u konzolu i njihovo momentalno izvršavanje

```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vpc>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello")
Hello
>>> _
```

# Prvi Python program

- Izvorni kod Python programa je u tekstualnom formatu
- Python fajlovi obično imaju ekstenziju **py**
- Kreirati fajl sa nazivom: **myprogram.py**<sup>1</sup>
- Sadržaj fajla treba da bude:  
`print("Hey people")`<sup>2</sup>
- Startovati fajl/program konzolnom komandom:  
`python myprogram.py`<sup>3</sup>
- Kada je Python program jednom kreiran, može se startovati neograničeno komandom:  
`python myprogram.py`



```
C:\> Command Prompt

C:\Users\vpc>echo print("Hey people!") > myprogram.py
C:\Users\vpc>python myprogram.py
Hey people!
C:\Users\vpc>
```

The screenshot shows a Windows Command Prompt window. The first command is `echo print("Hey people!") > myprogram.py`, which creates a file named `myprogram.py` containing the text `print("Hey people!")`. The second command is `python myprogram.py`, which runs the program and outputs `Hey people!`. Red numbers 1, 2, and 3 are placed above the file name, the code line, and the command line respectively, corresponding to the steps in the list.

# Vežba 1

---

- Pokušajte samostalno da napravite i startujete Python programe poput programa myprogram, samo sa drugačijom porukom
- Na primer

**Program:**

simpsons.py

**Poruka:**

Are we there yet?

**Program:**

fiddlesticks.py

**Poruka:**

Your bidding master!

**Program:**

calculator.py

**Poruka:**

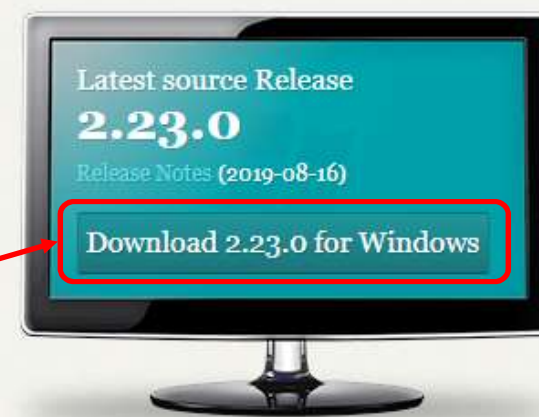
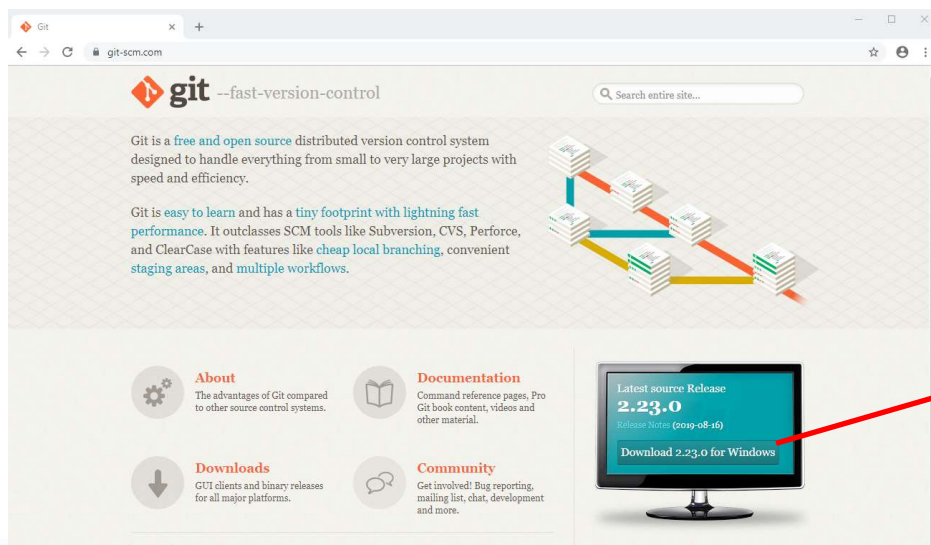
2+3

# Git i preuzimanje primera

- Jedan od najkorišćenijih alata za kolaboraciju je alat GIT
- GIT treba preuzeti sa sledeće adrese:

<https://git-scm.com/>

Na Linux i MacOS platformama GIT je obično preinstaliran, pa je samo potrebno potvrditi njegovo postojanje kucanjem komande **git** ili **git --version**.



# Git instalacija

Choosing the default editor used by Git

Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The [Vim editor](#), while powerful, [can be hard to use](#). Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

<https://gitforwindows.org/>

< Back

Next >

Cancel

Adjusting your PATH environment

How would you like to use Git from the command line?

☐ Use Git from Git Bash only

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ Git from the command line and also from 3rd-party software

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

☐ Use Git and optional Unix tools from the Command Prompt

Both Git and the optional Unix tools will be added to your PATH.  
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

< Back

Next >

Cancel

Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

☒ Use the OpenSSL library

Server certificates will be validated using the ca-bundle.crt file.

☐ Use the native Windows Secure Channel library

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

<https://gitforwindows.org/>

< Back

Next >

Cancel

Configuring the line ending conversions

How should Git treat line endings in text files?

☒ Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ('core.autocrlf' is set to 'true').

☐ Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ('core.autocrlf' is set to 'input').

☐ Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ('core.autocrlf' is set to 'false').

<https://gitforwindows.org/>

< Back

Next >

Cancel

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

☐ Use MinTTY (the default terminal of MSYS2)

Git Bash will use MinTTY as terminal emulator, which sports a resizable non-rectangular selections and a Unicode font. Windows console programs as interactive Python must be launched via "winpty" to work in MinTTY.

☒ Use Windows' default console window

Git will use the default console window of Windows ("cmd.exe"), which with Win32 console programs such as interactive Python or node.js, but very limited default scroll-back, needs to be configured to use a Unicode font to display non-ASCII characters correctly, and prior to Windows 10 the window was not freely resizable and it only allowed rectangular text selections.

<https://gitforwindows.org/>

< Back

Next >

Cancel

Configuring extra options

Which features would you like to enable?

☒ Enable file system caching

File system data will be read in bulk and cached in memory for certain operations ('core.fscache' is set to 'true'). This provides a significant performance boost.

☒ Enable Git Credential Manager

The [Git Credential Manager for Windows](#) provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).

☐ Enable symbolic links

Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

<https://gitforwindows.org/>

< Back

Next >

Cancel

Configuring experimental options

Which bleeding-edge features would you like to enable?

☐ Enable experimental, built-in add -i/-p

(NEW!) Use the experimental built-in interactive add ("git add -i" or "git add -p"). This makes it faster (especially the startup!), but it is not yet considered robust.

<https://gitforwindows.org/>

< Back

Install

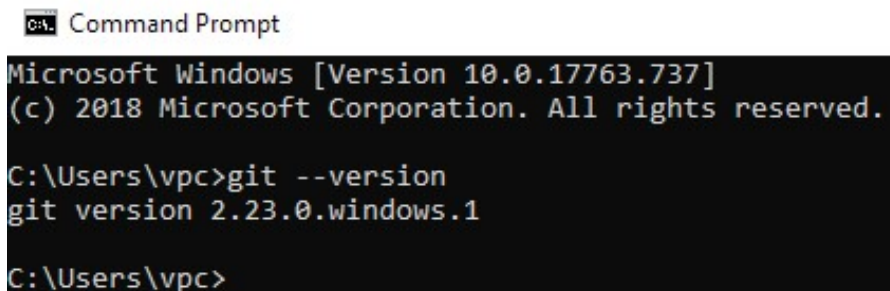
Cancel



# Git provera ispravnosti

---

- Da biste utvrdili funkcionalnost programa Git, kucajte u konzoli komandu **git** ili **git --version**



```
C:\> Command Prompt
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\vpc>git --version
git version 2.23.0.windows.1

C:\Users\vpc>
```

Ako je konzola bila otvorena tokom instalacije Git-a, git naredba neće biti u putanji do sledećeg otvaranja konzole



# Preuzimanje primera

- Primeri za ovaj predmet, nalaze se u repozitorijumu **python-ppf**.
- Za preuzimanje repozitorijuma, u bilo kom direktorijumu kucati naredbu:  
`git clone https://vmaric@bitbucket.org/vmaric/python-ppf.git`
- Nakon preuzimanja repozitorijuma, primeri sa slajdova će biti dostupni unutar istoimenih direktorijuma (na primer, primer **ppf-ex01** će se nalaziti u direktorijumu **python-ppf/ppf-ex01**, a primer **ppf-ex05 MyApp** će se nalaziti u direktorijumu: **python-ppf/ppf-ex05/MyApp**)

```
C:\Users\vpc>git clone https://vmaric@bitbucket.org/vmaric/python-ppf.git
Cloning into 'python-ppf'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.

C:\Users\vpc>cd python-ppf

C:\Users\vpc\python-ppf>
```

# Preuzimanje primera

---

- Pokušaj da pronađeš primere iz vežbe 1, znajući da je njihova oznaka: ppf-ex01

# Preuzimanje primera

```
C:\Users\vpc\python-ppf>cd ppf-ex01

C:\Users\vpc\python-ppf\ppf-ex01>dir
Volume in drive C has no label.
Volume Serial Number is BC47-F949

Directory of C:\Users\vpc\python-ppf\ppf-ex01

10/07/2019  12:12 PM    <DIR>          .
10/07/2019  12:12 PM    <DIR>          ..
10/07/2019  12:12 PM                12 calculator.py
10/07/2019  12:12 PM                31 fiddlesticks.py
10/07/2019  12:12 PM                28 simpsons.py
               3 File(s)                71 bytes
               2 Dir(s)  5,211,693,056 bytes free

C:\Users\vpc\python-ppf\ppf-ex01>python simpsons.py
Are we there yet?

C:\Users\vpc\python-ppf\ppf-ex01>_
```

Listu primera sa servera uvek možete ažurirati komandom: **git pull** unutar direktorijuma repozitorijuma

# Okruženje za rad

---



**PyCharm**

<https://www.jetbrains.com/pycharm/>



**Eclipse**

<https://www.eclipse.org/>



Visual Studio Code

**Visual Studio Code**

<https://code.visualstudio.com/>

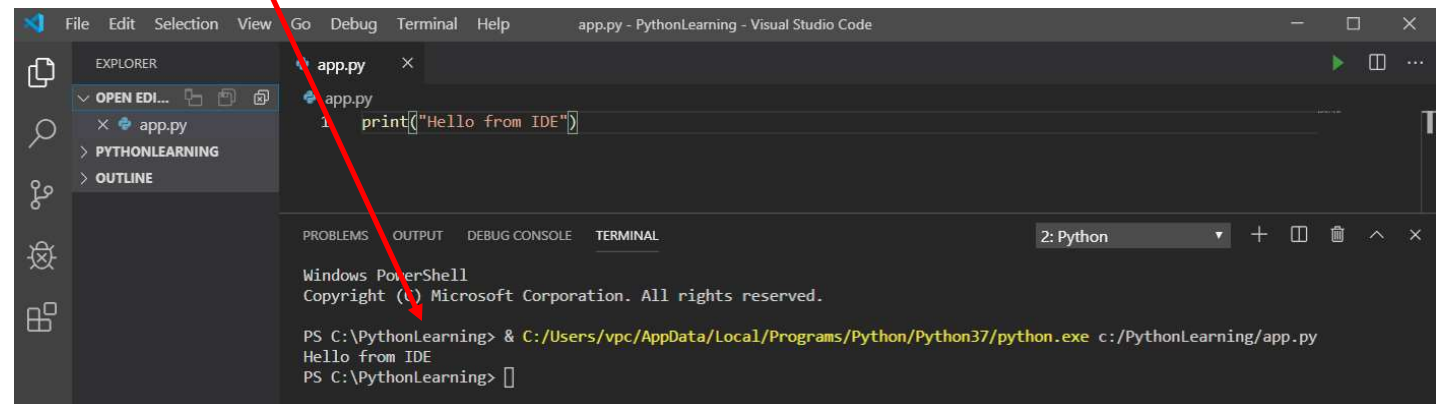
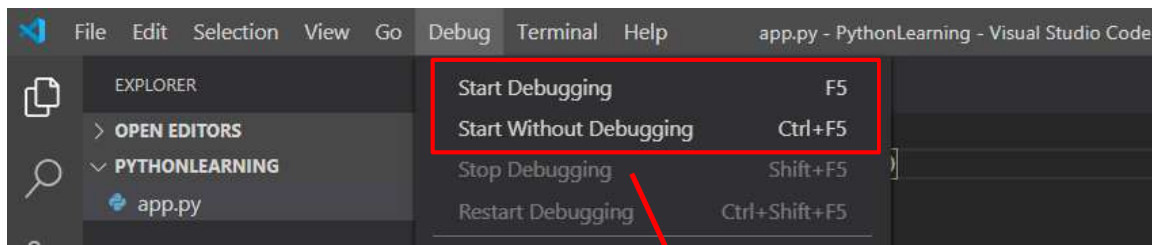
*Ovo ćemo koristiti  
na časovima*

# Okruženje za rad

Da bi Visual Studio Code uspešno radio sa Python-om, treba instalirati podršku za ovu platformu



Kreirani Python fajl se može startovati opcijama **Debug->Start Debugging** ili **Debug->Start Without Debugging**



# Python sintaksa i pravila kodiranja

---

# Sintaksa i semantika

---

- Sintaksa jezika podrazumeva upotrebu njegovih ključnih elemenata / reči
  - U formalnom jeziku možemo reći **dobar dan**, ali ne možemo reći **rdbaor nda**.
  - Na primer, fraza: **hello world**, ne može se pojaviti kao ključna reč Python programa, ali fraza **hello = 123** može
- Semantika jezika podrazumeva pravilno raspoređivanje, elemenata / ključnih reči jezika
  - U formalnom jeziku možemo reći **dobar dan**, ali ne možemo reći **dan dobar**.
  - U python-u možemo napisati **hello = 123**, ali ne možemo napisati **123 = hello**.



# Iskazi i linije

---

- U pythonu se svaka kompletna "misao" predstavlja linijom koda

**Ispravno je**

left edge

```
print("Start game!")
```

**Nije ispravno**

left edge

```
print("Start game!")
```

- Jedna linija koda je linija teksta, delimitirana novim redom

left edge

```
print("Start game!")  
print("The game has been started!")  
print("Game finished")
```

# Iskazi i linije

---

- Linije se takođe mogu razdvajati i oznakom ;

```
left edge print("Welcome");  
           print("To");  
           print("Python");
```

- Tada je moguće napisati i više programskih linija u jednom redu

```
left edge print("Welcome");print("To"); print("Python");
```

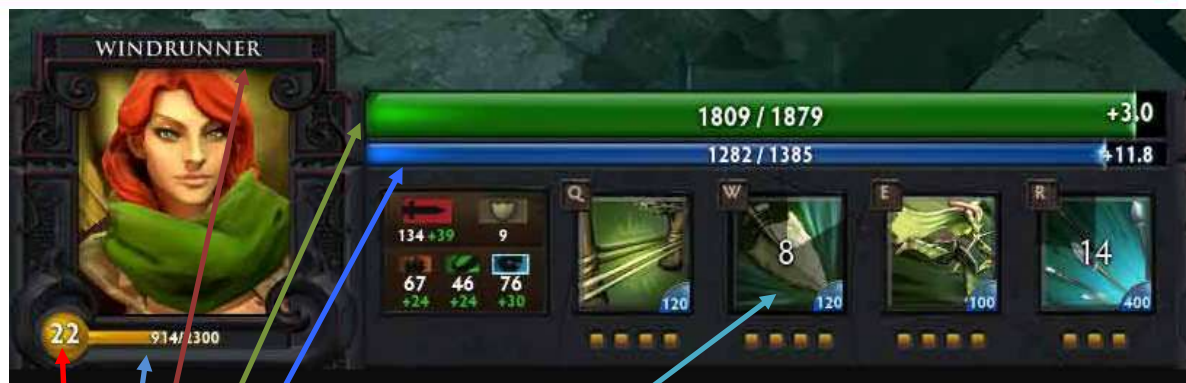
# Identifikatori

---

- Identifikatori su nazivi kojima imenujemo korisnički definisane elemente našeg programa
- To su najčešće **promenljive**, ali mogu biti i **funkcije** ili **klase**
- Kod imenovanja ovih elemenata, možemo koristiti **velika** i **mala slova**, bez specijalnih karaktera i spejsova.
- Brojeve možemo koristiti, ali ne na početku identifikatora
- Identifikatori takođe ne mogu biti **predefinisane ključne reči** jezika
- Osim pomenutog, identifikatori bi trebali (ali to nije sintaksno obavezno) da budu **intuitivno napisani**, jednim istim stilom

# Identifikatori

Dobri



Loši

level  
experience  
heroName  
health  
mana  
heroClass  
ability2Cooldown

lvl  
ex  
hero name  
h  
M@na  
class  
2abilityCooldown

# Blokovi koda

---

- Blokovi koda su tematski grupisane linije koda
- Blokovi koda se u Python-u označavaju uvlakom

```
if 1 == 1:  
    print("It's equal")  
print("Nah, it's not")
```

- Dužina uvlake nije bitna, sve dok su uvlake istog nivoa, iste dužine

```
if 1 == 1:  
    print("It's equal")  
print("Nah, it's not")
```

```
if 1 == 1:  
    print("It's equal")  
print("Nah, it's not")
```

# Tekst

---

- Tekstualni sadržaji se smeštaju u navodnike
- Navodnici mogu biti dvostruki ili jednostruki, ali se ne smeju kombinovati
- Otvoreni navodnici moraju biti i zatvoreni

## Ispravno

```
print("Hello")  
print('Hello')
```

## Neispravno

```
print("Hello'  
print("Hello)
```

# Prazan prostor

---

- Python ignoriše prazan prostor između redova, pa su sledeća dva koda ista sa stanovišta izvršavanja

```
print("You chosen")
```


```
print("well")
```

```
print("You chosen")  
print("well")
```

# Komentarisanje koda

- Osim praznih linija, Python interpreter takođe ignoriše i komentare u kodu
- Komentari koda se označavaju znakom **#** ispred komentara. Sve nakon ovog znaka će biti tretirano kao komentar, a ne kao program

```
print("You chosen")  
#print("well")
```



```
myprogram.py  
1 print("You chosen")  
2 #print("well")
```

OUTPUT    TERMINAL    DEBUG CONSOLE

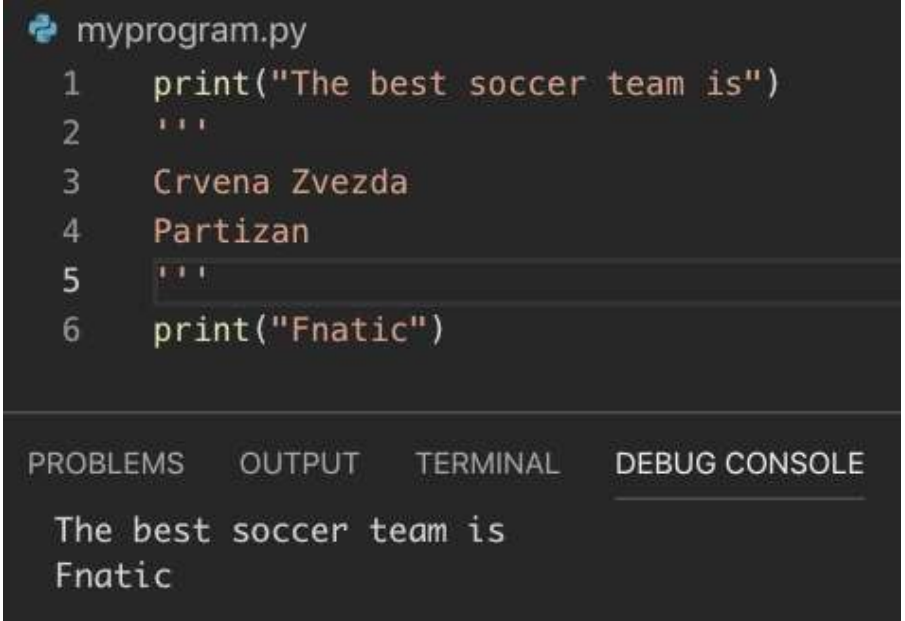
You chosen



# Višelinijnsko komentarisanje koda

- Za komentarisanje dužeg teksta, oznaka # ispred komentara nije dobro rešenje, jer se mora označavati svaka linija
- Tada se može koristiti oznaka ''' Na početku linije ispred komentara i na početku linije na kraju komentara

```
print("The best soccer team is")
'''
Crvena Zvezda
Partizan
'''
print("Fnatic")
```



The screenshot shows a code editor window titled 'myprogram.py'. The code contains a multi-line comment. Below the code, there are tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', and 'DEBUG CONSOLE'. The 'OUTPUT' tab is selected, showing the execution results.

```
myprogram.py
1  print("The best soccer team is")
2  '''
3  Crvena Zvezda
4  Partizan
5  '''
6  print("Fnatic")
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

The best soccer team is  
Fnatic

## Vežba 2 (ppf-ex02 war.py)

---

- Program ima sintaksnu grešku. Potrebno je ispraviti grešku i startovati program

```
war.py
1  print("War. War never changes")
2
```

## Vežba 3 (ppf-ex02 earth.py)

---

- Program ima sintaksnu grešku. Potrebno je ispraviti grešku i startovati program

```
1 print("The earth was nearly wiped clean of life.")
2 print("A great cleansing, an atomic spark struck by human hands, quickly raged out of control. Spears of
   nuclear fire rained from the skies.")
3 print("Continents were swallowed in flames and fell beneath the boiling oceans. Humanity was almost
   extinguished, their spirits becoming part of the background radiation that blanketed the earth.")
```

## Vežba 4 (ppf-ex02 setup.py)

- Program ima sintaksne greške. Potrebno je ispraviti greške i startovati program

```
1  '''
2  ~~~~~
3  One of the northern tribes claims they are descended from one such Vault. They hold that their founder and
4  ancestor, one known as the "Vault Dweller," once saved the world from a great evil. According to their
5  legend, this evil arose in the far south. It corrupted all it touched, twisting men inside, turning them
6  into beasts. Only through the bravery of this Vault Dweller was the evil destroyed.
7  But in so doing, he lost many of his friends and suffered greatly, sacrificing much of himself to save the
8  world. When at last he returned to the home he had fought so hard to protect, he was cast out. Exiled. In
9  confronting that which they feared, he had become something else in their eyes...and no longer their
10 champion.
11 ~~~~~
12 '''
13 print("Set your appearance")
```

# Vežba 5 (ppf-ex02 appearance.py)

---

- Program ima sintaksne greške. Potrebno je ispraviti greške i startovati program

```
1  '''  
2  Forsaken by his people, he strode into the wasteland. He traveled far to the north, until he came to the  
   great canyons. There, he founded a small village, Arroyo, where he lived out the rest of his years. And so,  
   for a generation since its founding, Arroyo has lived in peace, its canyons sheltering it from the  
   outside world. It is home. Your home.  
3  '''  
4  print("Set your appearance")  
5  z = input("Male or \"female\" (male/female)?")  
6  if z=="male":  
7      print("You choose male")  
8  else:  
9      print("You choose female")
```

# Vežba 6 (ppf-ex02 window.py)

- Program ima sintaksne greške. Potrebno je ispraviti greške i startovati program
- Ne treba analizirati šta program radi, niti ga razumeti. Već samo treba ispraviti sintaksne greške ukoliko postoje

```
1  from tkinter import *
2  import tkinter.messagebox
3
4  window = Tk()
5
6  window.title("Hello Python!!!")
7  window.resizable(False,False)
8  window.geometry('%dx%d+%d+%d' % (600, 400, (window.winfo_screenwidth()/2)-300,
9  window.winfo_screenheight()/2-200))
10
11 lbl = Label(window, text="START GAME")
12 lbl.place(x=300, y=200, anchor="center")
13 lbl.config(width=400)
14
15 def start(event):
16     tkinter.messagebox.showwarning('Sorry bro!',The game will start, as soon as you implement her',
17     parent=window)
18
19 lbl.bind('<Button-1>', start)
20
21 window.mainloop()
```